Process Model-Based Continuous Improvement of Election Process Quality and Robustness

Leon Osterweil[1], George Avrunin[1], Matt Bishop[2], Lori Clarke[1],
1: University of Massachusetts Amherst. 2: University of California Davis

Abstract

Demonstrate how applying software analysis to rigorously-defined models of processes can identify defects and vulnerabilities and lead to improvements in those processes. We use Model Checking to identify process defects and Fault Tree Analysis to show how incorrect performance (by humans or machines) creates opportunities for attacks. We also show how both analysis techniques can be combined to provide automated support for the synthesis of attacks and the subsequence verification of the robustness of the processes to such attacks.

Objective

A holistic approach for using rigorous analysis of precisely defined processes to incrementally improve the quality and robustness of a process.

- Highly automated
- Based on rigorously-defined process models
- Applies formal analysis techniques
- Supports continuous improvement of processes

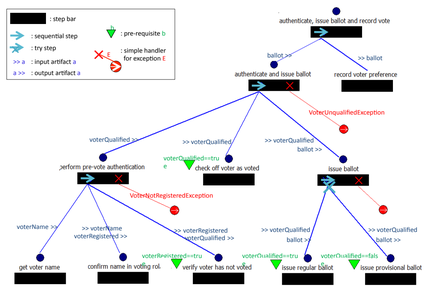
Preliminary Results

The approach was applied to the Yolo and Marin County, California, election processes:

- Modeled parts of the Yolo and Marin County election processes in Little-JIL
- Applied Fault Tree Analysis to identify process vulnerabilities that allow an unqualified voter to receive a regular ballot
- Identified process paths whose execution could violate desired election properties
- Modeled potential attack based on identified vulnerabilities
- Analyzed process robustness in presence of an attack

Process modeling using Little-JIL

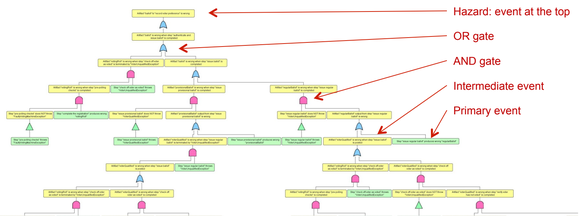
✓Well-defined semantics
✓Expressive
✓Accessible



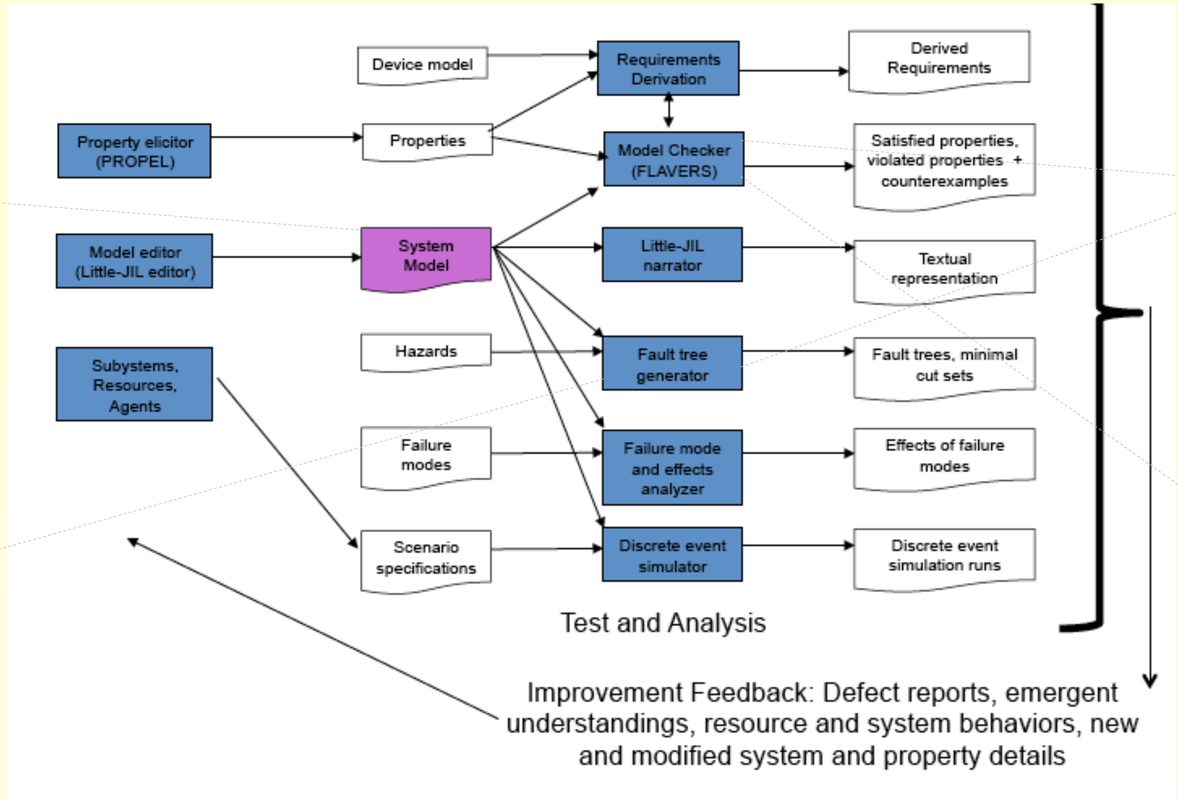
Model of part of Yolo County election process

Vulnerability identification using automated Fault Tree Analysis

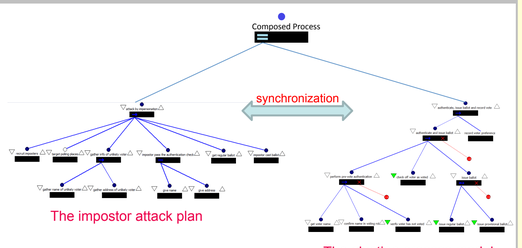
- A deductive, top-down analysis to find out which events in a system could lead to a given ***hazard***
- A ***fault tree*** is a graphical model of the various combinations of events producing the hazard
- A ***minimal cut set*** (***MCS***) is a minimal set of primary events all of whose occurrence ensures that the hazard event occurs. It indicates a system ***vulnerability*** that an adversary may be able to exploit to create the hazard



Top part of the Fault Tree derived from Yolo County election process



Process Model & Attack Plan Integration



Robustness evaluation using automated model checking

A formal verification technique that

- exhaustively explores all possible execution paths in a finite model of a process,
- determines whether a particular property holds in the model,
- produces a counterexample if the property does not hold.

Successful hazard defense is represented by the property: the attack cannot complete successfully

Future Directions

✓ Increase level of automation in:
- Attack plan construction from MCSs
- Attack plan integration with process model

✓ Improve derived fault trees
- Increase completeness of the fault tree derivation algorithm
- Improve hazard specification

✓ Improve process models
- Analyze more parts of election processes
- Verify more election process properties
- Ensure process models provide sufficient details for formal analyses

References

A Systematic Process-model-based Approach for Synthesizing Attacks and Evaluating Them *H. Phan, G. S. Avrunin, L. A. Clarke, L. J. Osterweil, M. Bishop,* 2012 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE '12), August 6-7, 2012
https://www.usenix.org/system/files/conference/evtwote12/evtwote12-final26.pdf

Modeling and Analyzing Faults to Improve Election Process Robustness
B. I. Simidchieva, S. J. Engle, M. Clifford, A. C. Jones, S. Peisert, M. Bishop, L. A. Clarke, L. J. Osterweil, 2010 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE '10), August 9-10, 2010, Washington, DC.

Contact: Lee Osterweil: ljo@cs.umass.edu

This research was supported by National Science Foundation (NSF) under Awards CCF-0905530, CNS-1049738, CCF-0820198, and IIS-0705772.